Comunicaciones en Estadística
Diciembre 2025, Vol. 18, No. 2, pp. 1–9

Filtro de SPAM basado en aprendizaje profundo

Deep learning-based SPAM filter

Felipe Algarra[a]
angambaa@unal.edu.co

Andres David Leon Hernandez[b]
aleonh@unal.edu.co

Juan Camilo Camargo[c]
jucamargop@unal.edu.co

Danna Lesley Cruz Reyes[d]
dlcruzr@unal.edu.co

Resumen

La clasificación de correos electrónicos como spam o no spam es un problema clásico en el procesamiento de lenguaje natural. En este trabajo se implementan dos enfoques de aprendizaje profundo para abordar esta tarea: un modelo basado en BERT y una red neuronal recurrente LSTM. Se comparan sus rendimientos en términos de precisión, recall, F1 score y eficiencia computacional. Ambos modelos se entrenaron y evaluaron sobre el Enron Email Corpus, alcanzando una exactitud global del 97 % y un F1 score equilibrado para ham y spam. El modelo BERT presenta una leve mejora en métricas de robustez, aunque implica mayores tiempos de entrenamiento e inferencia; por su parte, LSTM sigue siendo una solución efectiva cuando se diseña y entrena adecuadamente con un consumo de recursos sensiblemente menor. Estos hallazgos evidencian que, según los requisitos de latencia y capacidad de cómputo, es posible optar por un enfoque transformer de alto rendimiento o una implementación recurrente más ligera.

***Palabras clave*:** Clasificación de texto, aprendizaje profundo, spam, BERT, LSTM, PLN, redes neuronales..

Abstract

The classification of emails into spam or ham is a classic problem in natural language processing. In this study, we implement two deep learning approaches for tackling this task: a BERT-based model and a recurrent neural network using LSTM. Their performance is compared in terms of precision, recall, F1-score, and computational efficiency. Both models were trained and evaluated on the Enron Email Corpus, achieving an overall accuracy of 97 % and a balanced F1-score for both spam and ham. While the BERT model shows slightly improved robustness metrics, it also requires significantly more computational resources. Meanwhile, LSTM remains an effective solution when properly designed and trained with lower computational demand. These findings suggest that depending on latency and computing constraints, one may choose between a high-performance transformer-based approach or a more lightweight recurrent implementation.

***Keywords*:** Spam detection, Natural Language Processing, BERT, LSTM, Deep Learning, Text Classification, Transformers, RNN..

1. Introducción

El crecimiento del volumen de información textual digital ha hecho cada vez más urgente el desarrollo de sistemas automáticos para filtrar contenidos no deseados, como los correos electrónicos de tipo spam. Estos mensajes, además de ser molestos, pueden suponer riesgos de seguridad, como estafas, phishing y malware.

[a]Departamento de Estadística, Universidad Nacional de Colombia
[b]Departamento de Estadística, Universidad Nacional de Colombia
[c]Departamento de Estadística, Universidad Nacional de Colombia
[d]Departamento de Estadística, Universidad Nacional de Colombia

La detección automática de spam es una tarea fundamental en el área del procesamiento de lenguaje natural (NLP), y ha sido abordada tradicionalmente con técnicas de aprendizaje automático como Naive Bayes, máquinas de vectores de soporte (SVM) o árboles de decisión. Estas técnicas, aunque efectivas hasta cierto punto, presentan limitaciones al enfrentarse con textos más complejos o con spam sofisticado que imita el lenguaje humano de manera más natural.

El avance reciente de los modelos de aprendizaje profundo ha permitido desarrollar sistemas más potentes para tareas de clasificación de texto. En particular, las redes neuronales recurrentes (RNN), como las LSTM (Long Short-Term Memory), han demostrado ser efectivas en la captura de dependencias temporales en secuencias de texto, mejorando significativamente el rendimiento en tareas como análisis de sentimiento, traducción automática y clasificación de documentos. Por otro lado, los modelos basados en transformers, como BERT (Bidirectional Encoder Representations from Transformers), han revolucionado el NLP al permitir un mejor entendimiento del contexto bidireccional de las palabras.

Este proyecto tiene como objetivo implementar y comparar modelos BERT y LSTM para la clasificación de spam, evaluando su rendimiento y eficiencia. La hipótesis es que BERT, al incorporar contexto bidireccional, debería superar al modelo LSTM, aunque con mayor costo computacional.

2. Metodología

Este trabajo adopta un enfoque comparativo entre dos arquitecturas de redes neuronales profundas. La ruta de trabajo general incluye: preprocesamiento de datos, entrenamiento de modelos, evaluación con métricas y análisis comparativo de resultados.

El conjunto de datos utilizado proviene de Kaggle (Spam Mails Dataset) y está basado en el corpus de correos electrónicos del popular 'Enron Email Dataset'. Contiene 33716 correos electrónicos, de los cuales aproximadamente el 50.9 % están etiquetados como spam. Los datos incluyen el texto completo y el asunto del correo electrónico, así como su etiqueta correspondiente (spam o ham).

En este estudio se implementaron dos enfoques de aprendizaje profundo para la clasificación de correos electrónicos como spam o no spam: una red neuronal recurrente tipo LSTM y un modelo basado en transformadores, específicamente BERT. Ambos modelos fueron entrenados y evaluados sobre el Enron Email Corpus, ampliamente utilizado en la literatura como referencia estándar para tareas de detección de spam. A continuación, se describen detalladamente las arquitecturas y los fundamentos teóricos de cada modelo.

Las redes neuronales recurrentes (RNN) han sido utilizadas con éxito en tareas de procesamiento de lenguaje natural (PLN), debido a su capacidad para modelar secuencias. No obstante, presentan limitaciones al manejar dependencias de largo plazo. Para superar esta dificultad, se utiliza una arquitectura conocida como Long Short-Term Memory (LSTM), introducida por (Hochreiter and Schmidhuber, 1997), la cual incorpora puertas de entrada, olvido y salida que permiten preservar información relevante en secuencias extensas.

El modelo LSTM utilizado en este trabajo toma como entrada vectores de palabras generados mediante una capa de embedding, seguidos por una capa LSTM con unidades ocultas. La salida se pasa a través de una capa densa con activación sigmoide para obtener la probabilidad de que un mensaje sea spam.

Este tipo de arquitectura ha demostrado ser eficiente en escenarios donde se dispone de recursos computacionales limitados, y cuando el objetivo es construir modelos livianos con buena capacidad de generalización (Yin et al., 2017).

El segundo enfoque implementado corresponde a BERT (Bidirectional Encoder Representations from Transformers), propuesto por (Devlin et al., 2019), el cual representa un cambio de paradigma en las tareas de PLN. BERT se basa en una arquitectura de transformadores introducida por (Vaswani et al., 2017), que permite capturar dependencias contextuales bidireccionales en el texto, mejorando significativamente el desempeño en tareas de clasificación, respuesta a preguntas y análisis de sentimientos.

Para esta aplicación, se utilizó la versión preentrenada `bert-base-uncased`, posteriormente ajustada a la tarea específica de clasificación binaria. El preprocesamiento del texto se realizó con el tokenizador de BERT, y el modelo se entrenó utilizando el optimizador AdamW, con función de pérdida binaria y ajuste de pesos por clase para mitigar desbalance en los datos.

Aunque los modelos BERT suelen implicar un mayor costo computacional, ofrecen ventajas importantes en precisión y capacidad de generalización, particularmente en tareas que involucran lenguaje natural no estructurado o ambiguo.

3. Resultados

Para visualizar la distribución de la longitud de los mensajes, se aplicó un filtro que excluye aquellos con más de 5000 caracteres, ya que representaban valores atípicos que distorsionaban la escala del gráfico original. La Figura **??** corresponde al histograma filtrado que permite observar con mayor claridad el comportamiento general del conjunto de datos. Se observa que:

- La mayoría de los mensajes, tanto ham como spam, tienen una longitud inferior a los 1000 caracteres, concentrándose en el rango de 0 a 500.
- Los mensajes spam tienden ligeramente a ser más extensos que los ham.
- La distribución de ambos tipos de mensaje decrece conforme aumenta la longitud, lo que indica que los mensajes largos son menos frecuentes en el conjunto de datos.

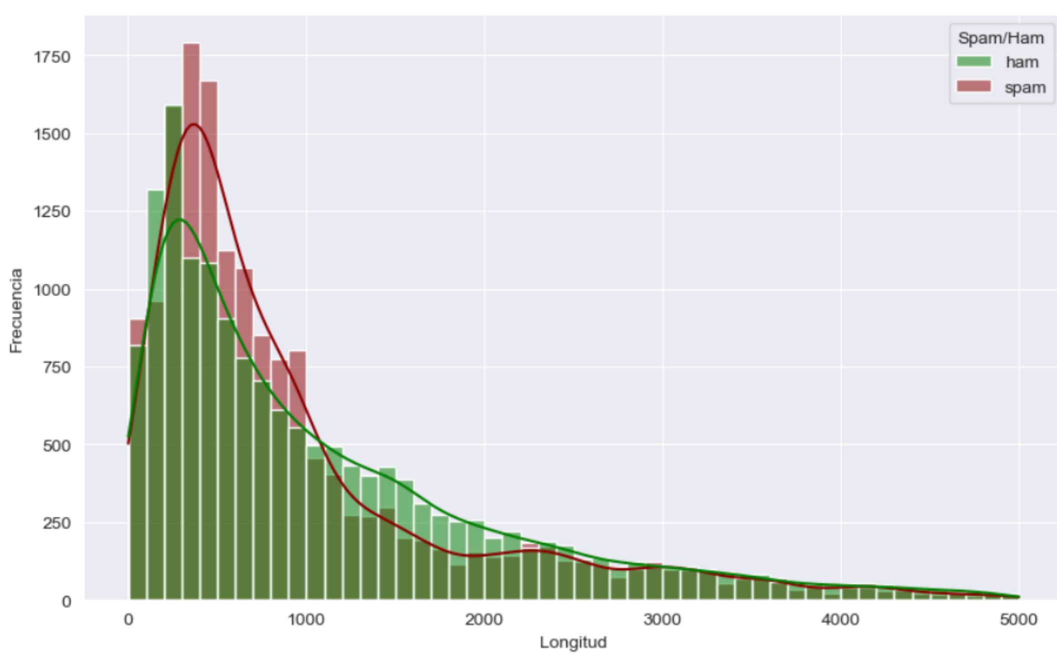

Figura 1: Distribución de la longitud de los mensajes

En la Tabla 1 se muestra la media y la mediana de la longitud de los mensajes para cada categoría, tras eliminar los outliers.

Aunque el histograma sugería que los mensajes spam tendían a ser más largos, las estadísticas descriptivas revelan lo contrario: los mensajes ham tienen en promedio mayor longitud que los spam. Esto puede deberse a la presencia de ciertos mensajes ham extensos que elevan la media. La diferencia también se refleja en la mediana, aunque de forma menos pronunciada.

Tabla 1: Estadísticas descriptivas longitud de los mensajes

Spam/Ham	Media	Mediana
ham	1093.55	741
spam	956.61	614

3.1. Modelo LSTM

Para abordar el problema de clasificación utilizando un modelo basado en redes neuronales de tipo LSTM (Long Short-Term Memory). Primero, se realizó un preprocesamiento de los datos que consistió en combinar los campos *Subject* y *Message* que constan del asunto y el contenido del correo respectivamente. Después, se hizo una limpieza del texto, en la cual, se eliminaron los caracteres no alfabéticos, se convirtió todo el contenido a minúsculas y se tokenizó utilizando herramientas de `nltk`. Posteriormente, se filtraron las stopwords del inglés y se aplicó un proceso de stemming con el algoritmo de Porter, con el objetivo de reducir las palabras a su forma base y estandarizar el texto. A partir del texto ya procesado, se creó un vocabulario limitado a las 10.000 palabras más frecuentes del conjunto de datos. Cada mensaje fue transformado en una secuencia de enteros, asignando un identificador único a cada palabra, y utilizando un valor especial para aquellas que no estaban incluidas en el vocabulario. Debido a que los mensajes tienen longitudes variables, se diseñó una función de ensamblado personalizada (collate function) que rellena las secuencias con ceros hasta alcanzar la longitud del mensaje más largo dentro de cada lote. Esto permitió que las secuencias fueran procesadas de manera eficiente por la red. Posteriormente, se dividió el conjunto de datos en una proporción de 80 % para el entrenamiento y 20 % para el testing. La arquitectura del modelo incluyó una capa de embeddings de dimensión 64, seguida de una capa LSTM con 128 unidades ocultas, y finalmente una capa completamente conectada con activación sigmoide para generar una salida binaria. El modelo fue entrenado durante 10 épocas utilizando la función de pérdida binaria BCELoss y el optimizador Adam, con una tasa de aprendizaje de 0.001.

3.2. Modelo BERT

Para abordar la clasificación de mensajes como spam o ham con un modelo basado en transformers de manera computacionalmente más eficiente, se dividió aleatoriamente el conjunto de datos en una muestra representativa del 20 % (aprox. 7000) del total de los mensajes del dataset, que a su vez se estratificó en un 80 % para entrenamiento y un 20 % para prueba. Para la arquitectura, se empleó `distilbert-base-uncased` adaptado a una salida binaria. Se entrenaron tres modelos BERT, cada uno con las combinaciones de hiperparámetros del Cuadro 2.

En todos los casos se implementó el optimizador AdamW junto a un scheduler lineal de learning rate sin warm up. En cada época se calculó la pérdida de entropía cruzada, se retropropagaron los gradientes y se actualizaron todos los pesos del modelo. Se evaluó la precisión del clasificador sobre el conjunto de prueba calculando predicciones con argmax sobre los logits. Se midieron precisión, recall, F1 score y exactitud, y se generó la matriz de confusión para analizar falsos positivos y falsos negativos. Las distintas configuraciones se compararon para elegir el balance óptimo entre rendimiento y costo computacional.

Para finalizar, se entrenaron nuevamente tres modelos BERT, con sus respectivos hiperparámetros (Cuadro 2), esta vez congelando las capas intermedias. Es decir, solo se estimaron los parámetros al final de la red relacionados con la clasificación. Dichos modelos se compararon con su contraparte entrenada por completo.

Tabla 2: Hiperparámetros de los modelos BERT entrenados

Modelo	Tamaño de batch	Épocas	Learning rate
1	16	2	5e-5
2	32	3	5e-5
3	32	2	1e-4

4. Resultados de Evaluación

Se evaluaron tres configuraciones del modelo BERT sobre el mismo conjunto de prueba (1288 mensajes), y se compararon con el modelo LSTM descrito anteriormente. A continuación se presentan las métricas principales y las matrices de confusión correspondientes.

Finalizado el entrenamiento, el modelo LSTM se evaluó con el conjunto de datos destinado para ello. Los resultados obtenidos fueron positivos, se alcanzó una precisión del 97 %, lo que refleja una buena capacidad del modelo para identificar correctamente los mensajes de spam y ham. La Figura 2 ilustra una matriz de confusión, la cual permitió observar con mayor detalle los errores cometidos por el modelo. Se encontró un número reducido de falsos positivos y falsos negativos, lo que sugiere que la red logró generalizar bien sin sobreajustarse.

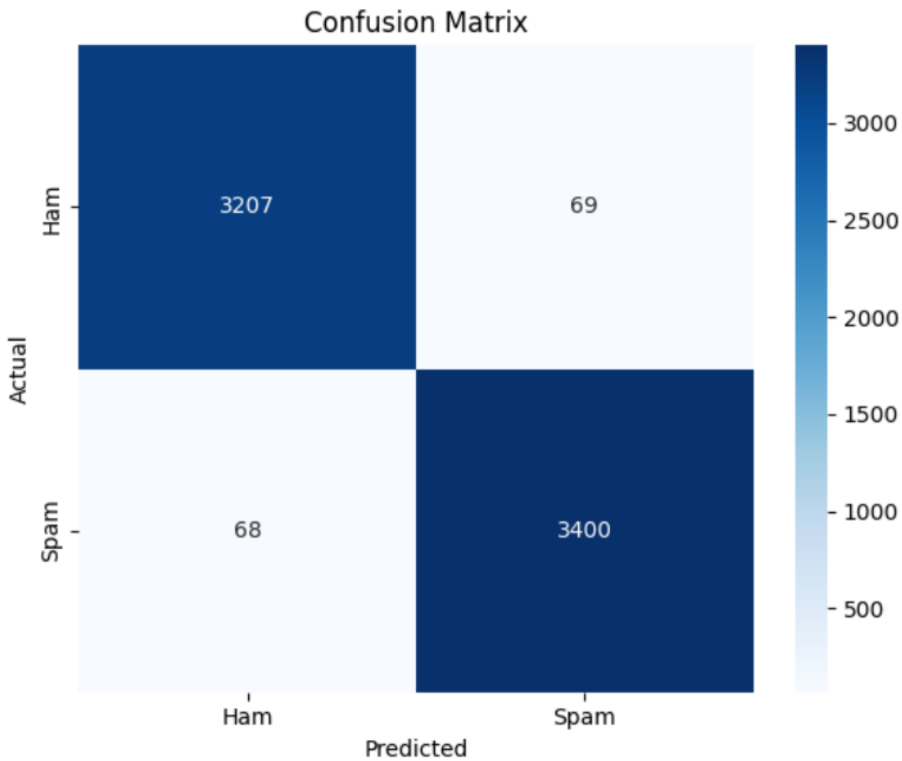

Figura 2: Matriz de confusión del modelo LSTM

En la primera configuración, el modelo obtuvo un recall de 1.00 en la clase ham, lo que indica que todos los mensajes ham (632) fueron correctamente identificados, pero alcanzó un recall de solo 0.28 en la clase spam. La precisión global fue de 0.63 y el F1 score ponderado de 0.58. La matriz de confusión en la Figura 3 muestra que 475 mensajes spam fueron clasificados erróneamente como ham, mientras que 629 ham se etiquetaron correctamente y 3 ham se confundieron como spam.

Tabla 3: Resultados de evaluación - Modelo 1

Clase	precision	recall	f1-score	support
Ham	0.57	1.00	0.72	632
Spam	0.98	0.28	0.43	656
accuracy			0.63	1288
macro avg	0.78	0.64	0.58	1288
weighted avg	0.78	0.63	0.58	1288

La segunda configuración mejoró significativamente el equilibrio entre clases, alcanzando precision, recall y F1 score de 0.97 para ham y spam, con una exactitud global de 0.97. La matriz de confusión en la Figura 4 registra 23 falsos positivos y 16 falsos negativos, lo que confirma un desempeño suficientemente uniforme en ambas categorías.

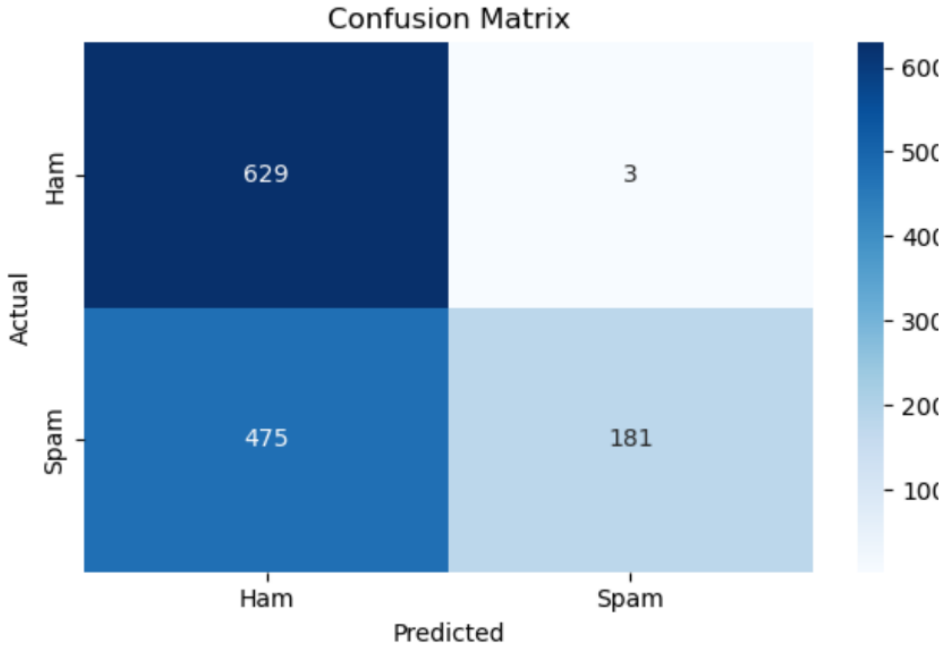

Figura 3: Matriz de confusión del modelo 1 de BERT

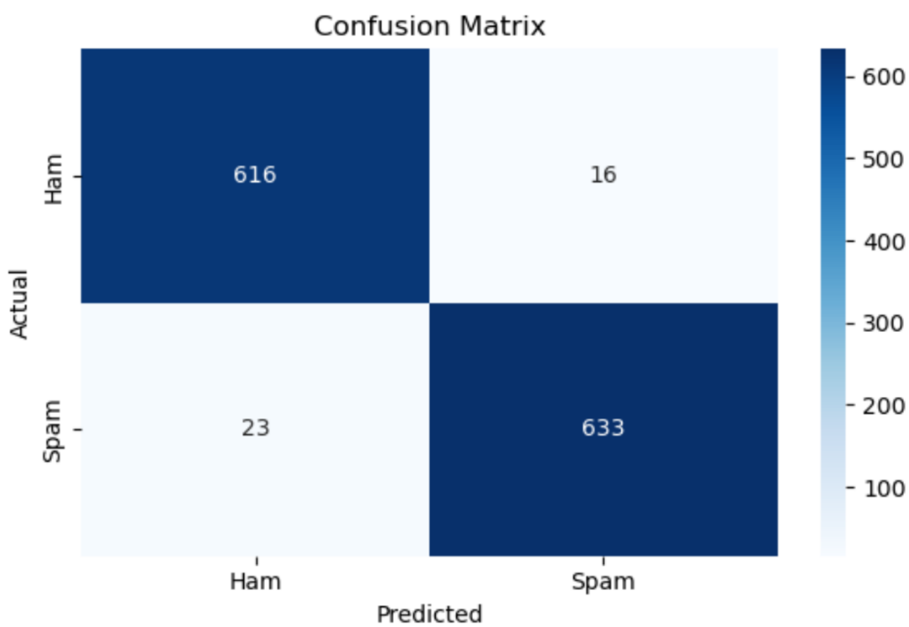

Figura 4: Matriz de confusión del modelo 2 de BERT

Tabla 4: Resultados de evaluación Modelo 2

Clase	precision	recall	f1-score	support
Ham	0.96	0.97	0.97	632
Spam	0.98	0.96	0.97	656
accuracy			0.97	1288
macro avg	0.97	0.97	0.97	1288
weighted avg	0.97	0.97	0.97	1288

En la tercera configuración, el modelo no detectó ningún mensaje spam, obteniendo un recall de 0.00 en esa clase y una exactitud global de 0.49 como se muestra en la Tabla 5.

Tabla 5: Resultados de evaluación Modelo 3

Clase	precision	recall	f1-score	support
Ham	0.49	1.00	0.66	632
Spam	0.00	0.00	0.00	656
accuracy			0.49	1288
macro avg	0.25	0.50	0.33	1288
weighted avg	0.24	0.49	0.32	1288

Todos los modelos BERT con capas congeladas presentaron un desempeño casi tan malo como el del modelo 3 de BERT sin congelar. Este fue un resultado inesperado, si bien la clasificación de spam no se piensa como un caso de uso típico de BERT, tampoco se piensa que este modelo tenga gran dificultad en encontrar los patrones que distinguen a un correo spam de uno ham tal como se observa en la Figura 5.

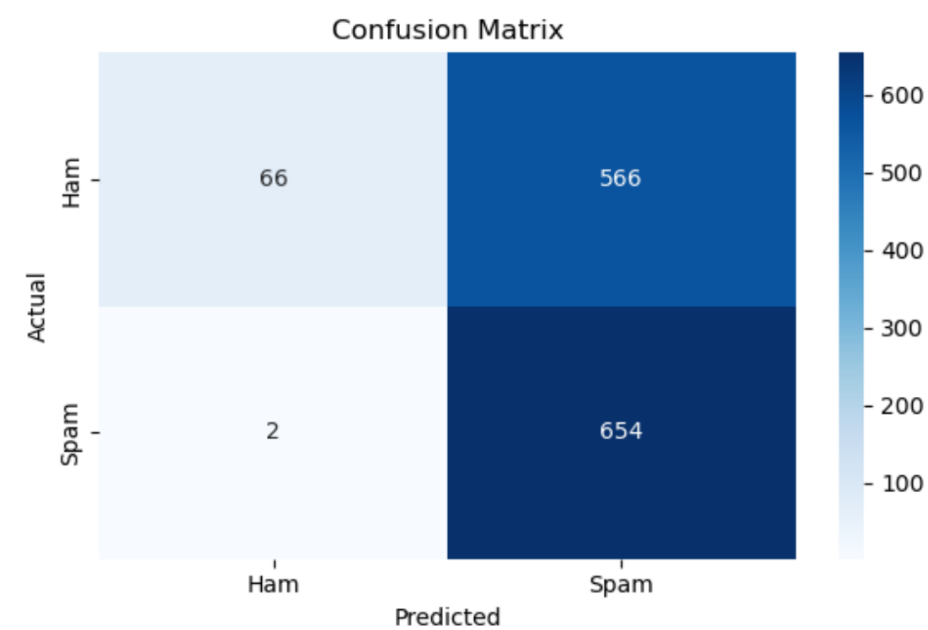

Figura 5: Matriz de confusión del modelo 2 de BERT congelado

En conjunto, el modelo 2 de BERT y el LSTM ofrecen rendimientos muy similares en términos de métricas de clasificación, pero mientras BERT requiere mucho más tiempo de entrenamiento e inferencia, el LSTM presenta una opción más eficiente computacionalmente sin sacrificar precisión.

5. Conclusiones

Los experimentos realizados permiten concluir que los modelos basados en transformers y en redes recurrentes ofrecen soluciones complementarias para la detección de spam en correos electrónicos. El BERT Modelo 2 demostró el mejor desempeño global, alcanzando una exactitud de 97 %, un F1 score equilibrado para ambas clases y un bajo número de falsos positivos y falsos negativos. Este resultado confirma que el contexto bidireccional y la capacidad de atención multicabeza de BERT aportan una ventaja significativa en la comprensión semántica de los mensajes. Sin embargo, dicho rendimiento superior se logra a costa de un mayor tiempo de entrenamiento e inferencia, así como de mayores requerimientos de memoria.

Por su parte, el modelo LSTM obtuvo una precisión igualmente alta del 97 % y mostró un comportamiento muy consistente entre clases, con un número reducido de errores de clasificación. Si bien el enfoque basado en LSTM no representa el estado del arte actual en procesamiento de lenguaje natural, los resultados demuestran que sigue siendo una solución efectiva cuando se diseña y entrena adecuadamente. Además, su arquitectura más sencilla y sus tiempos de cómputo inferiores lo convierten en una alternativa atractiva para entornos con recursos limitados o en aplicaciones que requieren respuestas en tiempo real.

En términos prácticos, la elección entre BERT y LSTM dependerá del balance deseado entre precisión y eficiencia computacional. Para sistemas donde la máxima exactitud es crítica y los recursos de cómputo están disponibles, BERT Modelo 2 es la opción recomendada. En cambio, cuando se prioriza la rapidez de entrenamiento e inferencia con hardware modesto, el modelo LSTM constituye una solución robusta y eficiente. Como líneas futuras, se sugiere explorar variantes ligeras de transformers (por ejemplo TinyBERT), combinaciones híbridas que integren embeddings contextuales con capas recurrentes, y técnicas avanzadas de tuning que optimicen aún más la relación entre desempeño y costo computacional.

Códigos

El código en R utilizado en este trabajo puede consultarse en:
`https://github.com/leon-andres/ClasificadorSPAM-NaiveBayes`.

Recibido: Agosto de 2025
Aceptado: Diciembre de 2025

Referencias

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 3 edition, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hobson Lane and Maria Dyshel. *Natural Language Processing in Action*. Manning Publications, 2 edition, 2025.

Richard McElreath. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC Press, 2 edition, 2020.

Simon JD Prince. *Understanding Deep Learning*. MIT Press, 2023. URL `https://udlbook.github.io/udlbook`.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. A comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. *Dive into Deep Learning*. Cambridge University Press, 2023. URL `https://d2l.ai/`.